



XMPP

XEP-0287: Spim Markers and Reports

Evgeniy Khramtsov

<mailto:ekhramtsov@process-one.net>

<xmpp:xram@jabber.ru>

2010-10-04

Version 0.1

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This document defines an XMPP protocol extension that enables XMPP entities to interact with spim filters by marking unsolicited or suspicious XMPP stanzas.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1	Introduction	1
2	Requirements	1
3	Glossary	2
4	Use Cases	2
4.1	Spim Marker	2
4.2	Spim Report	3
5	Business Rules	4
6	Determining Support	5
7	Security Considerations	5
7.1	CAPTCHA challenges	5
7.2	Fake <mark/> element	6
7.3	Fake <report/> element	6
7.4	Multiple fake <report/> elements	6
7.4.1	Single filtering entity	6
7.4.2	Several filtering entities	6
7.5	Fake IQ-set report	6
8	IANA Considerations	7
9	XMPP Registrar Considerations	7
9.1	Protocol Namespaces	7
10	XML Schema	7
10.1	urn:xmpp:spim-marker:0	7
10.2	urn:xmpp:spim-report:0	8
11	Acknowledgements	9

1 Introduction

There are various spim protection methods exist in XMPP: [Privacy Lists](#)¹, [CAPTCHA Forms](#)², [Simple Communications Blocking](#)³, [Incident Reporting](#)⁴ and [Entity Reputation](#)⁵. But they may not be sufficient enough:

- [Privacy Lists](#)⁶ and [Simple Communications Blocking](#)⁷ define blocking mechanism only which is not always appropriate.
- [CAPTCHA Forms](#)⁸ interacts badly with automated software such as gateways.
- [Incident Reporting](#)⁹ implies trusted network of servers.
- [Entity Reputation](#)¹⁰ concentrates on ranking only.

Service administrators might want to deploy server-based spim recognition software to fill in the gaps. However, every automated spim recognition suffers from false positives - situations where a stanza incorrectly qualified as spim. To avoid them, a spim filter doesn't block suspicious stanza, but marks it and sends to a client in a regular manner. A client software doesn't need to interrupt a user when processing such marked stanzas: for example, it may put them silently in "SPAM" folder, so a user can look through them at any time later. Furthermore, a spim filter may take user's experience into account. When a user receives an unsolicited stanza, he or she can mark it as spim. In this case a client software sends an automatic complaint to a server-based spim filter. This specification deals with both cases. Thus, in contrast to [SPIM-Blocking Control](#)¹¹, it doesn't introduce any spim blocking techniques. Also, the various spim recognition procedures that may be employed by the server are beyond the scope of this document.

2 Requirements

An implementation compliant with this document MUST support spim markers as described in [Spim Marker](#) use case. Support for spim reports, as described in [Spim Report](#) use case, is RECOMMENDED.

¹XEP-0016: Privacy Lists <<http://xmpp.org/extensions/xep-0016.html>>.

²XEP-0158: CAPTCHA Forms <<http://xmpp.org/extensions/xep-0158.html>>.

³XEP-0191: Simple Communications Blocking <<http://xmpp.org/extensions/xep-0191.html>>.

⁴XEP-0268: Incident Reporting <<http://xmpp.org/extensions/xep-0268.html>>.

⁵XEP-0275: Entity Reputation <<http://xmpp.org/extensions/xep-0275.html>>.

⁶XEP-0016: Privacy Lists <<http://xmpp.org/extensions/xep-0016.html>>.

⁷XEP-0191: Simple Communications Blocking <<http://xmpp.org/extensions/xep-0191.html>>.

⁸XEP-0158: CAPTCHA Forms <<http://xmpp.org/extensions/xep-0158.html>>.

⁹XEP-0268: Incident Reporting <<http://xmpp.org/extensions/xep-0268.html>>.

¹⁰XEP-0275: Entity Reputation <<http://xmpp.org/extensions/xep-0275.html>>.

¹¹XEP-0159: SPIM-Blocking Control <<http://xmpp.org/extensions/xep-0159.html>>.

3 Glossary

The following terms are used throughout this document:

Filtering Entity An XMPP entity which performs spim recognitions, blocks or marks suspicious stanzas and accepts spim reports. Example: a server or an external component with built-in spim recognition module.

Receiving Entity An XMPP entity which directly receives marked stanzas and sends spim reports. Example: a client or a conference (Multi-User Chat XEP-0045: Multi-User Chat <<http://xmpp.org/extensions/xep-0045.html>>.).

4 Use Cases

4.1 Spim Marker

The filtering entity marks abusive stanza by adding <mark/> child element qualified by the 'urn:xmpp:spim-marker:0' namespace. The element MUST possess the 'filter' attribute whose value MUST be a full jid of the filtering entity. The <mark/> element MAY contain character data which SHOULD be a human-readable description of the reason to mark. The filtering entity MUST NOT add more than one <mark/> element and MUST delete all other <mark/> elements matching itself before adding a new one. The filtering entity MAY remove any <mark/> elements matching itself even if it doesn't add a new one.

Listing 1: User's Server Marked Abusive Message

```
<message from='robot@abuser.com/zombie'
  to='innocent@victim.com/laptop'
  id='spam1'>
  <body>Love pills - 75% OFF</body>
  <mark xmlns='urn:xmpp:spim-marker:0'
    filter='victim.com'/>
    Unsolicited advertising
  </mark>
</message>
```

Listing 2: Several Services Marked Abusive Message

```
<message from='robot@abuser.com/zombie'
  to='innocent@victim.com/laptop'
  id='spam1'>
  <subject>You won $1,000,000!</subject>
  <body>Visit http://www.abuser.com/</body>
  <mark xmlns='urn:xmpp:spim-marker:0'
    filter='dnsbl-filter.victim.com'>
    Blocked by too many DNSBLs
```

```

</mark>
<mark xmlns='urn:xmpp:spim-marker:0'
      filter='bayes-filter.victim.com' />
</message>

```

Processing rules of marked stanzas taken by the receiving entity are beyond the scope of this document. One possible solution is to put such stanzas silently in so-called "SPAM" folder.

4.2 Spim Report

If the filtering entity wishes to receive abuse report for the stanza, it MUST add <report/> child element qualified by the 'urn:xmpp:spim-report:0' namespace and MUST possess the 'key' and the 'filter' attributes. A value of the 'key' attribute is arbitrary, but SHOULD have at least 128 bits of randomness. The 'key' attribute is needed to match the corresponding complaint (if any) with the sender. The value of the 'filter' attribute MUST be a full jid of the filtering entity. The filtering entity MUST NOT add more than one <report/> element and MUST delete all other <report/> elements matching itself before adding a new one. The filtering entity MAY remove any <report/> elements matching itself even if it doesn't add a new one.

Listing 3: Multiple Filters Wishes to Receive Abuse Report

```

<presence type='subscribe'
          from='robot@abuser.com'
          to='innocent@victim.com'
          id='spam2'>
  <report xmlns='urn:xmpp:spim-report:0'
        key='571c9641d8442920'
        filter='filter.victim.com' />
  <report xmlns='urn:xmpp:spim-report:0'
        key='b258acbc4bb8e66ac'
        filter='victim.com' />
</presence>

```

The receiving entity MAY complain by sending an IQ-set containing the <query/> child element qualified by the 'urn:xmpp:spim-report:0' namespace. A value of the 'filter' attribute MUST be copied in the 'to' attribute of the IQ-set stanza. The element MUST possess 'key' attribute copied from the original stanza.

The receiving entity MUST ignore any <report/> elements generated by untrusted filtering entities. If there are more than one <report/> element matching the same filtering entity, all of them MUST be ignored.

Listing 4: Receiver Sends Complaint

```

<iq type='set'
    from='innocent@victim.com/laptop'
    to='filter.victim.com'

```

```

    id='complaint1'>
    <query xmlns='urn:xmpp:spim-report:0'
        key='571c9641d8442920' />
</iq>

<iq type='set'
    from='innocent@victim.com/laptop'
    to='victim.com'
    id='complaint2'>
    <query xmlns='urn:xmpp:spim-report:0'
        key='b258acbc4bb8e66ac' />
</iq>

```

The filtering entity MUST respond with an empty IQ-result stanza upon successful completion of the request:

Listing 5: Complaint Was Accepted

```

<iq type='result'
    from='filter.victim.com'
    to='innocent@victim.com/laptop'
    id='complaint1' />

<iq type='result'
    from='victim.com'
    to='innocent@victim.com/laptop'
    id='complaint2' />

```

5 Business Rules

A filtering entity SHOULD only add <mark/> or <report/> elements and a receiving entity SHOULD only process those elements if the corresponding stanza involves an interaction with a human user: subscription requests, messages, conference invites, voice calls, etc. For example, it doesn't make a lot of sense to mark [Software Information](#)¹² stanzas.

To avoid obvious false positives and user confusions, a filtering entity SHOULD NOT add <mark/> or <report/> elements to a stanza and a receiving entity SHOULD ignore <mark/> and <report/> elements of a stanza if:

- The receiving entity has the sender's subscription information of the type "both", "from" or "to".
- The receiving entity has pending subscription to the sender, i.e. subscription of type "none" and ask='subscribe'.
- The receiving entity has sent direct presence to the sender.

¹²XEP-0232: Software Information <<http://xmpp.org/extensions/xep-0232.html>>.

6 Determining Support

If an entity supports the spim markers, it MUST report that by including a service discovery feature of "urn:xmpp:spim-marker:0" in response to a [Service Discovery](#)¹³ information request. If an entity supports the spim reports, it MUST report that by including a service discovery feature of "urn:xmpp:spim-report:0" in response to a [Service Discovery](#)¹⁴ information request:

Listing 6: Service Discovery Information Request

```
<iq type='get'
  from='juliet@capulet.lit/balcony'
  to='capulet.lit'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 7: Service Discovery Information Response

```
<iq type='result'
  from='capulet.lit'
  to='juliet@capulet.lit/balcony'
  id='disco1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:spim-marker:0' />
    <feature var='urn:xmpp:spim-report:0' />
    ...
  </query>
</iq>
```

7 Security Considerations

7.1 CAPTCHA challenges

Care should be taken if a receiving entity chooses to generate a CAPTCHA challenge ([CAPTCHA Forms](#)¹⁵) in response to a marked stanza. A spim recognition system rarely has more than 5-10% of false positives. Thus, producing CAPTCHA images or audio/video samples is likely a waste of system resources and also may overload the receiving entity at high rate of spim stanzas.

¹³XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

¹⁴XEP-0030: Service Discovery <<http://xmpp.org/extensions/xep-0030.html>>.

¹⁵XEP-0158: CAPTCHA Forms <<http://xmpp.org/extensions/xep-0158.html>>.

7.2 Fake <mark/> element

A rogue server may add fake <mark/> elements to compromise filtering entities: a user may decide to remove such entities from the trusted list because, for example, he or she thinks they produce too many false positives. To avoid such situation, a filtering entity **MUST** remove any <mark/> elements matching itself before adding new <mark/> element as described in [Spim Marker](#) use case. Also, a filtering entity **MAY** remove any <mark/> elements matching itself even if it doesn't add a new one.

7.3 Fake <report/> element

An attacker may add fake <report/> element. For example, it may do that for checking an activity of the user. To avoid such situation, a receiving entity **MUST** send spim reports to the trusted filtering entities only as described in [Spim Report](#) use case.

7.4 Multiple fake <report/> elements

7.4.1 Single filtering entity

An attacker may add thousands of fake <report/> elements matching the single trusted filtering entity in one stanza. A poorly written receiving entity may generate a complaint for all of them. As an effect, a distributed DoS attack on the filtering entity is performed if there are multiple receiving entities involved. To avoid such situation, a receiving entity **MUST** ignore multiple <report/> elements matching the same filtering entity as described in [Spim Report](#) use case.

In its turn, a filtering entity **MUST** remove any <report/> elements matching itself before adding new <report/> element as described in [Spim Report](#) use case. Thus, it is guaranteed that the element will not be ignored by the receiving entity.

7.4.2 Several filtering entities

An attacker may gain an information about user's trusted filtering entities. In this case he or she may add the <report/> element per every such entity in one stanza. If there are too many filtering entities in the list, a user may generate enormous traffic when generating spim reports. Although this attack is not very effective, a client software **MUST** not generate spim reports without user's acknowledgement.

7.5 Fake IQ-set report

An attacker may try to mark an innocent user as a spimmer by producing several IQ-set stanzas qualified by "urn:xmpp:spim-report:0" containing different value of the 'key' attribute each

(so-called "dictionary attack"). As a protection, sanity checks MUST be performed when processing such reports. For example, if a filtering entity doesn't store any information about a receiving entity, the value of the 'key' attribute SHOULD have at least 128 bits of randomness.

8 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#) ¹⁶.

9 XMPP Registrar Considerations

9.1 Protocol Namespaces

This specification defines the following XML namespaces:

- urn:xmpp:spim-marker:0
- urn:xmpp:spim-report:0

Upon advancement of this specification from a status of Experimental to a status of Draft, the [XMPP Registrar](#) ¹⁷ shall add the foregoing namespace to the registry located at <http://xmpp.org/registrar/namespaces.html>, as described in Section 4 of [XMPP Registrar Function](#) ¹⁸.

10 XML Schema

10.1 urn:xmpp:spim-marker:0

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:spim-marker:0'
  xmlns='urn:xmpp:spim-marker:0'
```

¹⁶The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

¹⁷The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <http://xmpp.org/registrar/>.

¹⁸XEP-0053: XMPP Registrar Function <http://xmpp.org/extensions/xep-0053.html>.

```

    elementFormDefault='qualified'>

<xs:annotation>
  <xs:documentation>
    The protocol documented by this schema is defined in
    XEP-xxxx: http://www.xmpp.org/extensions/xep-xxxx.html
  </xs:documentation>
</xs:annotation>

<xs:element name='mark'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='xs:string'>
        <xs:attribute
          name='filter'
          type='xs:string'
          use='required' />
        <xs:attribute
          name='reason'
          type='xs:string'
          use='optional' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

</xs:schema>

```

10.2 urn:xmpp:spim-report:0

```

<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:spim-report:0'
  xmlns='urn:xmpp:spim-report:0'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-xxxx: http://www.xmpp.org/extensions/xep-xxxx.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:simpleContent>

```

```
<xs:extension base='xs:string'>
  <xs:attribute
    name='key'
    type='xs:string'
    use='required' />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

<xs:element name='report'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='xs:string'>
        <xs:attribute
          name='filter'
          type='xs:string'
          use='required' />
        <xs:attribute
          name='key'
          type='xs:string'
          use='required' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

</xs:schema>
```

11 Acknowledgements

Thanks to Sergei Golovan for the feedback.